IPST-CNAM
Programmation JAVA
NFA 002
Mercredi 27 Juin 2012

Avec document
Durée : **2 h30**
Enseignant : LAFORGUE Jacques

1ère Session NFA 002

L'examen se déroule en deux parties. Une première partie de 1h15mn, sans document, consacrée à des questions de cours, et une deuxième partie, avec document, consacrée en la réalisation de programmes Java.

Au bout de 1h15mn, les copies de la première partie seront ramassées avant de commencer la deuxième partie.

Pour la première partie, vous devez rendre le QCM rempli et les réponses aux questions libres écrites sur des copies vierges.

Pour la deuxième partie, vous écrivez vos programmes sur des copies vierges. Vous devez écrire les codes commentés en Java.

1ère PARTIE : COURS (sans document)

---

1. QCM (35 points)

Mode d'emploi :

Ce sujet est un QCM dont les questions sont de 3 natures :

- **les questions à 2 propositions**: dans ce cas une seule des 2 propositions est bonne.
  - +1 pour la réponse bonne
  - –1 pour la réponse fausse
- **les questions à 3 propositions** dont 1 seule proposition est bonne
  - + 1 pour la réponse bonne
  - –½ pour chaque réponse fausse
- **les questions à 3 propositions** dont 1 seule proposition est fausse
  - + ½ pour chaque réponse bonne
  - –1 pour la réponse fausse

Il s'agit de faire une croix dans les cases de droite en face des propositions.

On peut remarquer que cocher toutes les propositions d'une question revient à ne rien cocher du tout (égal à 0).

Si vous devez raturer une croix, faites-le correctement afin qu'il n'y ait aucune ambiguïté.

N'oubliez pas d'inscrire en en-tête du QCM, votre nom et prénom.

Vous avez droit à **4 points** négatifs sans pénalité.

NOM:	PRENOM:

En Java, la classe Hashtable<K,V> permet de gérer une collection d'éléments dont l'accès se fait par une donnée, appelée Key, et non par un rang comme cela est le cas dans la classe ArrayList		Q 1
1	OUI	
2	NON	

En Java, la classe Collections permet de trier les éléments de n'importe quelle collection (classe qui implémente l'interface List) grâce à la méthode ***sort***. Cette méthode fonctionne si la classe d'appartenance des éléments de la collection implémente l'interface :		Q 2
1	Comparator	
2	Comparable	
3	Compiler	

L'héritage multiple de classe est possible en Java		Q 3
1	OUI	
2	NON	

Une classe abstraite est une classe dont les types des attributs ne sont pas connus. C'est lors de l'instanciation de l'objet que les types des attributs sont résolus		Q 4
1	OUI	
2	NON	

Soit le code suivant : — Q 5

```
public class A extends B implements C
{
   private int attr_A;
   public A() {
      attr_A = 10; attr_B = "TOTO"; C.attr_C = 100; }
}

attr_B est un attribut protected de B
attr_C est un attribut static public de l'interface C
```

Ce code est correct

1	OUI	
2	NON	

Soit deux classes B et C qui héritent d'une classe abstraite A. Les classes B et C peuvent utiliser par héritage les attributs protected de la classe A.		Q 6
1	OUI	
2	NON	

Une classe qui contient au moins une méthode abstraite doit être déclarée abstraite.		Q 7
1	OUI	
2	NON	

En Java, on déclare un tableau qui contient des éléments dont la classe d'appartenance est une classe abstraite C.		Q 8
1	Cela n'est pas possible	
2	Cela est possible et on ajoute dans le tableau des objets de type C (t[i]=new C();)	
3	Cela est possible et on ajoute dans le tableau des objets de type B (t[i]=new B();) et la classe B hérite de C	

En JAVA, une classe peut implémenter plusieurs interfaces		Q 9
1	OUI	
2	NON	

En JAVA, une interface peut contenir des attributs :		Q 10
1	statics	
2	non statics	
3	privés	

En JAVA, une interface permet de :		Q 11
1	passer en paramètre d'une méthode un objet dont la classe d'appartenance implémente l'interface	
2	définir des traitements génériques sur des collections polymorphes	
3	créer des classes abstraites	

La déclaration d'une méthode suivante : `public void traitement(String s) throws MyException` précise que la méthode doit traiter l'exception MyException dans le corps de sa méthode.		Q 12
1	OUI	
2	NON	

Soit le code suivant qui ajoute un Individu dans un tableau : (Q 13)

```
public void ajouter(String nom) throws Exception
{
   Individu ind=null;
   try {
      ind = rechercher(nom);
   }
   catch(NonTrouveException ex) {
      tab[n++] = ind; }
}

La méthode rechercher retourne l'exception NonTrouveException si
le nom de l'individu n'est pas trouvé.
```

Soit le code suivant qui ajoute un Individu dans un tableau (code ci-dessus)		Q 13
1	si l'individu que l'on veut ajouter n'est pas trouvé alors la méthode retourne l'exception NonTrouveException	
2	si l'individu que l'on veut ajouter n'est pas trouvé et le tableau n'est pas plein alors l'individu est ajouté au tableau	
3	Si l'individu que l'on veut ajouter n'est pas trouvé et le tableau est plein alors la méthode retourne une exception prédéfinie Java (IndexOutOfBoundsException)	

Soit le code suivant :		Q 14

```
try{
   System.out.println("AAA");
   call();
   System.out.println("BBB");
}
catch(MyException ex) {
   System.out.println("DDD");
}
catch(Exception ex) {
   System.out.println("CCC");
}
```

avec la méthode ***call*** qui déclenche l'exception ***MyException.***

Ce code affiche :

1	AAA BBB	
2	AAA DDD CCC	
3	AAA DDD	

En JAVA, un package est un regroupement de classe Java (.java et .class). Ce regroupement (création du package) se fait avec la commande de compilation (option -d) : **javac -d <repertoire de regroupement> *.java**		Q 15
1	OUI	
2	NON	

Un design pattern est un modèle de description qui représente souvent une architecture de classes et dont l'objectif est de rendre réutilisable des choix de conception		Q 16
1	OUI	
2	NON	

Le design pattern Factory est un modèle de conception de la mise en œuvre		Q 17
1	d'une communication client serveur	
2	d'une fabrique d'objets décrits par une interface commune	
3	de la mise en facteur des méthodes de différentes classes dans une même interface	

Le Singleton est un design pattern qui		Q 18
1	est une classe contenant une méthode static qui crée un objet s'il n'est pas déjà créé et le retourne, sinon le retourne	
2	de créer un nouvel objet à chaque fois que l'on appelle la méthode static getInstance mais chacun des objets implémente une interface unique	
3	permet d'obtenir un objet qui est toujours le même. Il est unique dans la JVM d'exécution.	

Le code suivant crée un fichier de nom "exemple.txt" dans le répertoire courant d'exécution. Le fichier est créé, vide de toute information		Q 19

```
File fichier;
fichier = new File("exemple.txt");
```

1	OUI	
2	NON	

En JAVA, la gestion des entrées sorties se fait notamment par les classes qui se trouvent dans le package java.io		Q 20
1	OUI	
2	NON	

On a le code suivant :
`File fichier = new File("ListeDouble.bin);`
`FileOutputStream fos = new FileOutputStream(fichier);`
`DataOutputStream dos = new DataOutputStream(fos);`
`dos.writeInt(tab.length);`
`for(int i=0;i< tab.length;i++) dos.writeDouble( tab[i]);`
`dos.close();`
		Q 21

1	Ce code crée un fichier de nom 'ListeDouble.bin' contenant la liste de doubles	
2	Ce code crée un fichier dont les informations sont dans un format texte	
3	Ce code crée un fichier dont les informations sont dans un format binaire	

La class File ne gère que les fichiers. Pour gérer des répertoires on utilise la classe Directory		Q 22
1	OUI	
2	NON	

La sérialisation est un service du langage Java qui permet d'écrire et de lire n'importe quel objet dans un fichier binaire (Serializable) ou un fichier texte (XML)		Q 23
1	OUI	
2	NON	

Le code suivant est correct : Q 24

```
import java.io.*;

public class Terminal{
    static BufferedReader in =
        new BufferedReader(new InputStreamReader(System.in));

    public static String lireString() throws IOException
    {
        return in.readLine();
    }
}
```

1	OUI	
2	NON	

Le package est une unité de programmation permettant de regrouper et architecturer les classes du langage Java (prédéfinies ou développées) dans des répertoires et accessibles aux autres unités de programmation (programme Java, Applet, …).		Q 25
1	OUI	
2	NON	

Un répertoire de package est une chemin d'accès vers un répertoire qui contient les classes du package et ce répertoire ne peut pas contenir d'autres répertoires.		Q 26
1	OUI	
2	NON	

Soit le fichier suivant C:\CodeJava\exercices\cnam\util\Terminal.java. Le fichier Terminal.java contient en 1ère ligne : package exercices.cnam.util Le répertoire C:\bin est vide. On réalise la commande suivante : javac -d C:\bin C:\CodeJava\exercices\cnam\util\Terminal.java On obtient le résultat suivant : C:\bin\ exercices\cnam\util\Terminal.class		Q 27
1	OUI	
2	NON	

En JAVA, un thread est une JVM qui s'exécute en parallèle de la JVM dans laquelle le thread a été créé.		Q 28
1	OUI	
2	NON	

Pour créer et démarrer un thread, il faut :		Q 29
1	créer un objet dont la classe d'appartenance hérite de la classe Runnable et implémente la méthode run	
2	créer un objet dont la classe d'appartenance hérite de la classe Thread, implémente la méthode run et appeler la méthode start	

La classe A hérite de B qui hérite de C. C est une classe abstraite qui implémente une interface I. A et B ne sont pas des classes abstraites		Q 30
1	C peut implémenter une partie des méthodes de l'interface I	
2	B doit implémenter toues les méthodes de I qui n'ont pas été implémentées par C	
3	A ne peut pas implémenter des méthodes de l'interface I	

L'interface permet de :		Q 31
1	gérer des collections polymorphes (les éléments sont de type d'une interface)	
2	gérer les fichiers (interface avec le système d'exploitation)	
3	de passer en paramètre d'une méthode "un traitement" (traitements génériques)	

Un tableau Java peut contenir des type primitifs, peut contenir des objets et la classe d'appartenance de ces objets peut être une classe abstraite		Q 32
1	OUI	
2	NON	

Soit une collection "liste" définie par la classe ArrayList<Individu>. Nous proposons de vouloir trier les éléments de cette liste suivant 3 critères de tri différents. Pour réaliser ces 3 méthodes de tri différentes, il faut notamment :		Q 33
1	que la classe Individu implémente 3 interfaces différentes (1 interface par tri)	
2	créer 3 classes différentes Comparator1, Comparator2 et Comparator3 qui implémentent chacune l'interface Comparator<Individu>	
3	pour chacun des tris faire les appels : Collections.sort(liste, comparator1) Collections.sort(liste, comparator2) ou Collections.sort(liste, comparator3) où comparator1, comparator2, comparator3 sont des instances des classe Comparator1, Comparator2, Comparator3 qui implémentent chacune l'interface Comparator<Individu>	

En JAVA, la classe RuntimeException qui hérite de la classe Exception permet de déclencher une exception qui interrompt proprement et définitivement le runtime de la JVM.		Q 34
1	OUI	
2	NON	

La sérialisation a un prix: la compatibilité binaire des informations écrites qui dépendent des versions JAVA et surtout de la stabilité des classes dont les objets sont écrits sur ce principe		Q 35
1	OUI	
2	NON	

2. Questions libres (15 points)

Chaque question est notée sur 5 points.
*Vous répondez à ces questions sur une **copie vierge** en mettant bien le numéro de la question, sans oublier votre nom et prénom.*

Q 1

Expliquez à quoi servent les exceptions.

Q 2

Quels sont les composants (ou couches) d'une application Java basée sur les principes du modèle MVC.
Quel est le rôle de chacun de ces composants ?

Q 3

En programmation objet, donner une définition du polymorphisme.
Quels sont les deux moyens, en JAVA, pour mettre en œuvre le polymorphisme ? Commentez.

(Tourner la page)

2ème PARTIE : PROGRAMMATION (avec document)

Probleme [30 points]

On se propose de gérer un agenda (classe Agenda) de rendez-vous.
La classe Agenda gère une collection de RendezVous qui contient les méthodes :
- public void ajouter(RendezVous rdv)
- public void afficher()
- public void sauver()
- public void charger()

La méthode ***ajouter*** permet d'ajouter un rendez-vous de telle manière que les rendez-vous sont toujours ordonnés par ordre croissant sur la date puis sur l'heure de début.

La méthode ***afficher*** affiche à l'écran le contenu de l'agenda.

La méthode ***sauver*** sauve dans un fichier texte tout le contenu de l'agenda

La méthode ***charger*** initialise l'agenda depuis un fichier texte

Le nom du fichier est une constante de la classe Agenda.
La classe RendezVous doit contenir les méthodes de lecture et d'écriture d'un rendez-vous dans un flot de fichier texte.

La classe RendezVous est caractérisée par :
- la date du rendez-vous qui est une chaîne au format AAAAMMJJ (20120627)
- l'heure de début du rendez-vous qui est une chaîne au format HHhMM (ex: 18h05)
- l'heure de fin du rendez-vous qui est une chaîne au format HHhMM (ex: 21h00)
- le texte libre du rendez-vous (ex: "Examen NFA002")

Coder en JAVA les classes **Agenda** et **RendezVous**.

(Fin du sujet)